

FACE DETECTION AND RECOGNITION USING GOOGLE-NET ARCHITECTURE

Basma S. Mahmmed¹, Ammar I Majeed²

^{1,2} College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
Basmalshaker@gmail.com¹, ammar.alkindy@nahrainuniv.edu.iq²

Corresponding Author: Ammar I Majeed

Received:16/9/2022; Revised: 30/10/2022; Accepted: 04/12/2022

DOI:[10.31987/ijict.6.1.228](https://doi.org/10.31987/ijict.6.1.228)

Abstract- Face detection in secure places is an important application merging with machine vision. This paper presents a system for performing face detection and recognition using the existing architecture of Google-Net and transfer learning to make the network learn images based on pre-trained architecture. The design of the network leads to an architecture that maximizes the system's accuracy, and accurately detects faces that are saved to the database, and specifies the effect of the weights that were used within the nodes of the hidden layer, which is considered to be the most time-consuming task within the architecture. The main characteristics are explained, and the architecture and data set have been explained in detail with the design of a network, which provides an architecture that leads to maximizing the accuracy of the system, accurately detecting faces that are saved to the database, and specifying the effect of weights used within the nodes of the hidden layer, which is considered the most time-consuming task within the architecture. Experimental results show that using epochs of 10 and 100 samples implies 98.37% training accuracy, whereas other epochs either provide less accuracy or consume more time, and the number of the epochs and training samples can be modified according to the system requirements. Other factors like illumination, the color of the background, and face rotation or scaling were discussed as impact factors.

keywords: Face detection, face recognition, Google-Net, transfer learning

I. INTRODUCTION

Various authentication techniques are suggested for securing systems and data because user information might readily leak. Anyone must be able to "declare that they are who they claim to be," according to the authentication approach. Identity authorization for anything is irrelevant to the authentication process. Identity may also be stolen or revealed by close friends or coworkers. A few systems allow users to be identified by an object like an ID card or passport. It is useful for its intended use and relatively simple to steal or copy. As a result, biometrics, which are memorable and distinctive, have been considered as a choice for a secure process. A face, fingerprint, palm print, iris, or even hand geometry could be biometric technology. Face image appears to be one of such technologies' most widely employed characteristics because of its collectability, usability, and acceptability. Face recognition technology can be defined as a non-intrusive technology in terms of collectability and usability because it doesn't involve direct contact with test subjects. Numerous works assert that their correct recognition rates are greater than 95% in acceptability. Other biometrics, including fingerprints, also have a high matching accuracy rate. Yet, it is possible to create a fake fingerprint in three dimensions using a plastic sheet.

Additionally, for the recognition to be accurate, the fingerprint images stored in a database must be of high quality, which consumes a lot of space in the database when applied to a system with a large number of users. In contrast, iris scanning represents the most secure and reliable biometric technology; however, iris recognition equipment is pricey [1].

CNNs are a widely utilized and recognized approach for image classification. There are several CNN types, and while they all have the same basic structure, their topologies could vary. All the CNNs share identical fundamental structures, consisting of three different types of layers-pooling, convolutional, output, and fully connected-that are used by all CNNs. The convolutional layers contain some kernels, each of which calculates feature representations of the image. Feature maps represent feature representation, including the values for the face. A pooling layer is utilized after that to lower the feature maps' resolution. There are typically some pooling and convolutional layers. To sum up, the first convolution layer, which is also referred to as the low-level features, is used to detect the edges in the image. The following convolution layer is responsible for extracting additional abstract features after that. The fully connected (FC) layers often come after pooling and convolutional layers. The FC layers' task is to establish a communication channel between neurons. The output layer is the last layer of CNN. One function that could be applied in the final layer is the softmax function, which produces the final output. Using Fig. 1, a CNN architecture has been depicted [12] [13].

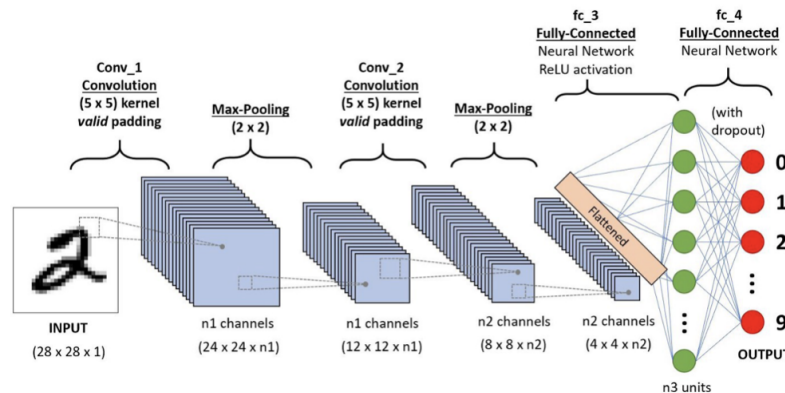


Figure 1: CNN architecture [14]

The goal of transfer learning (TL) is to apply knowledge that has been obtained from solving one problem to another that is unrelated but still present. Consider when a system needs to identify human faces in an image. Pre-trained models also known as models that have already been trained on many faces can be used in order to solve related problems without requiring a significant amount of data [15].

II. RELATED WORK

Face recognition and detection were investigated in many studies. Below is a list of some of these researchers, along with brief explanations of their studies: Taigman, Yaniv, et al. (2015) suggested a standard face recognition pipeline with 4 phases: detect \Rightarrow align \Rightarrow represent, \Rightarrow classify. They revisit both the representation and alignment steps by using explicit 3D face modeling to apply piecewise affine transformation and generate face representation from a 9layer deep NN. [2].

Yi Sun et al. (2015), the authors suggested a system that utilizes the impact of face verification and identification supervisory signals on the deep feature representation to coincide with the two components of creating the ideal features

for face recognition, i.e., decrease intra-personal variations and increase inter-personal variations. The combination of the two supervisory signals results in considerably better features than either [3].

Sun, Yi, et al. (2016), DeepID3 is the name of two extremely deep NN designs that the authors suggest using for face recognition. Stacked inception and convolution layers from Google-Net and VGG-net are combined to make such two architectures adequate for face recognition [4].

Schroff, Florian, et al. (2016) have suggested a technique that, as opposed to other DL methods, utilizes a deep CNN that is trained to directly improve the actual embedding. They trained with triplets of roughly aligned non-matching and matching face patches that have been produced by a new online triplet mining technique [5]. Qi, Xianbiao, and Lei Zhang, 2019, suggested a straightforward yet efficient centralized coordinate learning (CCL) approach that forces the features to be dispersedly spanned in the coordinate space while guaranteeing that the classification vectors lie on a hypersphere. The authors jointly formulate the learning of classification vectors and facial features [6].

Wang, Pin, Peng Wang, and En Fan 2021, used depth features and artificial features in order to extract the spatiotemporal properties of a video using a CNN and merged them with the features of the trajectory. Two techniques were developed to address the issue of low resolution preventing face images in surveillance video from being properly identified: the SPP-based CNN model and the multi-foot input CNN model [7].

Du, Hang, et al., 2021, the authors examine Near-infrared to visible (NIR-VIS) face recognition, which seeks to match a set of face images that have been taken from 2 separate modalities, as the most prevalent case in heterogeneous face recognition. While current DL-based approaches for NIR-VIS face recognition have made notable progress, they are facing some new challenges because of the COVID-19 pandemic, where individuals are advised to wear facial masks to stop the virus' transmission. The masked face in the NIR probe image makes this task, which we designate as NIR-VIS masked face recognition, difficult [8].

Li, Zheng, Xuemei Lei, and Shuang Liu, in 2022, constructed a lightweight NN that only needs a few weight representations and inexpensive operators. It could be used in embedded systems. A total of six convolutional layers make up the lightweight NN that was created in this study. Pooling has been achieved via convolution with a step size of two, and the batch normalization algorithm is employed in order to normalize the NN input [9].

In two key ways, Priya, R. L., et al. (2021) suggested a face recognition-based attendance system. A face-recognition algorithm can be trained over a set of images representing all the identities to identify when a new image belongs to one of these identities or is that of a stranger [10].

Sabu M. Thampi et al., 2022, put forth a technique that employs face recognition technology for machine aid throughout search operations and could be applied to the crucial development of applications that utilize CCTV footage from a camera network to locate a lost individual. In their method, they employ face recognition using one-shot learning to locate stranded individuals in large crowds [11]. Many other researchers employed AI methods in networks [16], IoT [17], the metaverse [18], and driverless cars [19]. Table I summarizes the related work.

TABLE I
Related Work Comparisons.

Reference	DEEP MODEL	DATASET	ACCURACY
2	DNN	face recognition	92.5%
3	VGG net, GoogLeNet	face recognition	96.0%
4	sparse ConvNets	face recognition	98%
5	deep convolutional network	LFW face recognition	99.63%
6	deep neural network (DNN), centralized coordinate learning (CCL)	face recognition	99.4674
7	CNN	mnist face recognition, COREL1000 dataset	98%
8	Near-infrared to visible (NIR-VIS)	face recognition	83.90%
9	deep neural network	MFLOPs face recognition	98.37%
10	One-Shot Learning	face recognition	97%
11	CNN	Children face recognition	95%
Proposed	CNN	face recognition	98%

III. PROPOSED SYSTEM

To design a network that provides an architecture that maximizes the system's accuracy, accurately detects faces that are saved into the database, and specifies the effect of weights used within the nodes of the hidden layer, which is considered the most time-consuming task within the architecture. The existing pre-trained or well-trained classification algorithm based on CNN can accurately classify images to different classes with high accuracy, which will reflect one limitation if the network works with data that has not been seen before, which it cannot classify. For this purpose, extra data should be added to the database. Within any system of identification and authorization using face detection, it is important to build a database of images that will be used in the identification process. For the system, accurate preprocessing steps are used to crop the face only, improve the image quality by equalizing the histogram, and remove noise so the image will be ready for feature extraction. The feature extraction phase and classification will be done internally inside the convolutional neural network. In this work, the Google-Net face recognition method is used to identify the faces within the database, with some modifications to the original architecture, as shown in Fig. 2.

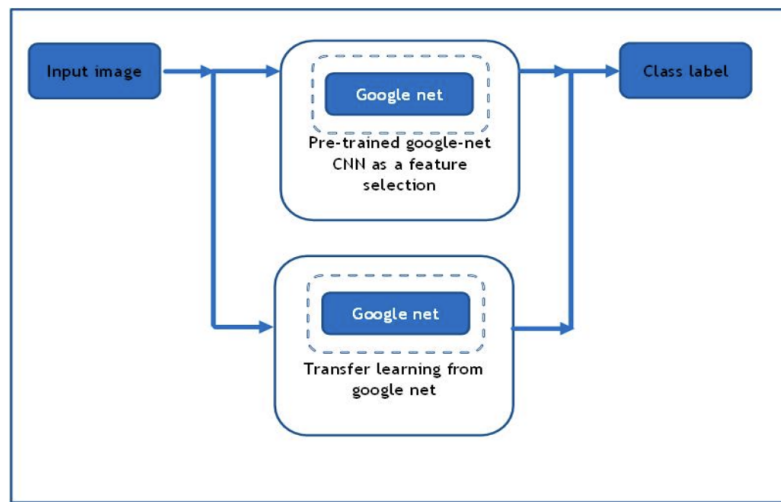


Figure 2: general process of the proposed system

Algorithm 1 face recognition using CNN
Input: images
Output: label images
Stage 1: preprocessing, where the input data set images are prepared for further processing Step1: The online captured image is resized to the proper unique size of the images, which is suitable for the design of our system. Step2: Noise removal Step3: Histogram equalization Step4: The image is converted to grayscale images to reduce the general number of the features that have been obtained. Stage2: face representation where the pre-trained GOOGLE-NET convolutional neural network was used Step5: Extraction of image features. Step6: Utilization of the extracted features. Stage3: classification and class label prediction

The work is done on the different datasets as explained in the data set section, and the system's main steps are shown in algorithm 1 . In addition, Fig. 3 depicts the general proposed system.

The suggested system architecture is depicted in Fig. 4.

IV. PREPROCESSING

The first stage of the proposed system is the preprocessing stage, where the input data set images are prepared for further processing. This stage is considered the most important step before the feature extraction since the behavior of the feature extractor changes according to the quality and main characteristics of the input image or frame, which should be clear, have the minimum level of noise, and have a properly defined size that helps to extract strong features. This stage is mainly split into four main steps:

- 1) Image resizing
- 2) Noise removal.

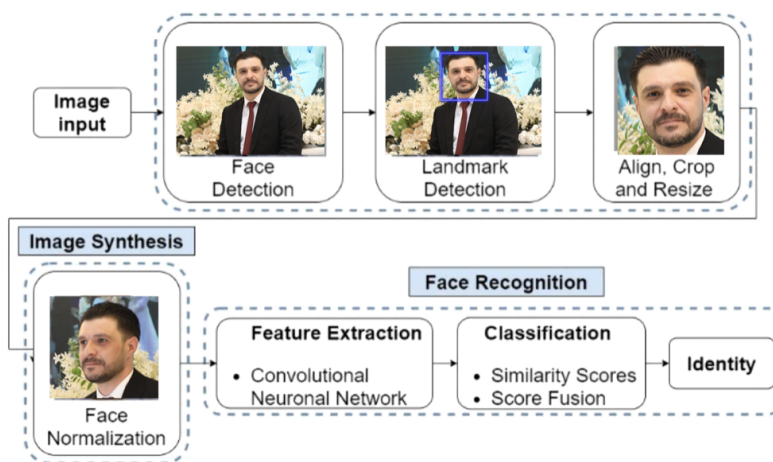


Figure 3: general system overview

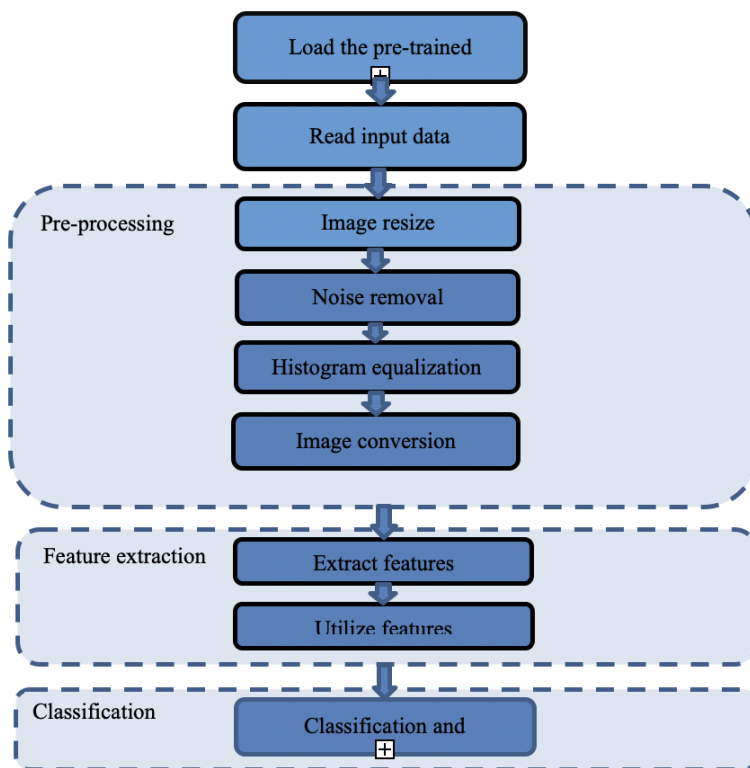


Figure 4: Proposed system architecture

- 3) Histogram equalization.
- 4) Image conversion.

A. Image Resize

The process of resizing an image is considered a critical situation in image processing when the image is not the unique size that the system desires and the number and quality of extracted features vary from image to image. With image resizing, each pixel of the image is updated simultaneously. This process will ignore all the unnecessary pixels if the image size is reduced, and if it enlarges, then new pixels should be added, and the pixels that are added were estimated. Within the proposed work, only one operation was applied. Mainly, the size was adjusted to 224*224 (the downsizing operation), which is suitable for the design of our system, as shown in Fig. 5.

B. Noise Removal

This step includes reducing the noise of the image to minimal levels, as shown in Fig. 6, where a median noise filter has been used to remove noise by smoothing the image, reducing or removing the visibility of noise in an image since this noise is mainly generated for different reasons such as camera lens, heat, movement, and dust. The importance of the noise removal step in the preprocessing stage within the proposed system is to remove the wrong or unwanted data (e.g., noise) before representing these data as features that may affect the system's accuracy.

C. Histogram equalization

This method is used to improve the quality of image contrast. It accomplishes this through the effective spreading out of the most frequent values of intensity, i.e., stretching out the image intensity range. Histogram equalization can be considered a part of the preprocessing stage. Preprocessing the image is needed before CNN trains or tests it. Preprocessing aims to reduce the computational cost, provide a faster ID system, and keep sufficient data for face representation.

D. GOOGLE-NET architecture

Google-Net can be described as a 22-layer deep CNN that is an Inception Network variant, a deep CNN developed by the researchers at Google. It is utilized for other computer vision tasks like adversarial training, face detection, recognition, etc. The Google-Net architecture consists of 22 layers (a total of 27 layers, including the pooling layers), and part of these layers are a total of 9 inception modules Fig. 7. The known and famous model was used to build the model, with modifications to the original specifications. Table II shows the original specification of the layers used for the Google-Net model.

The proposed design's first convolution layer has a filter (patch) size of 7×7 , making it significantly larger compared with other patch sizes in the network. The main goal of this layer is to instantly decrease the input image while preserving spatial information by using large filter sizes. A greater number of feature maps are created, yet the input image size (width and height) is decreased by a factor of 4 at the 2nd Conv layer and by a factor of 8 prior to reaching the first inception module. As a result of dimensionality reduction, the second cover layer uses the 1×1 Conv block and has a depth of

TABLE II
 GOOGLE-NET architecture.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

2 . Through 1×1 Conv blocks, dimensionality reduction enables the decrease of processing load by the reduction of the number of operations that are needed for every one of the layers. There are two max-pooling layers between the parts of the inception modules in the suggested system, which has nine inception modules. The input is down sampled as it is passed through the network by such max-pooling layers, which is accomplished by reducing the input data's width and height. Another efficient way of lowering the computational load on a network is by decreasing the input size between the inception modules. The input width and height are lowered to 1×1 , and the average pooling layer averages all feature maps that the previous inception module has created. The linear layer is utilized immediately prior to a dropout layer (40% dropout). The dropout layer can be defined as a regularization method utilized throughout the training to stop the network from overfitting. The dropout reduces the number of interconnected neurons in a NN at random. Each one of the neurons has a chance to be excluded from, or rather dropped from, the combined contribution from related neurons at each training stage. The linear layer has 1000 hidden units, or the same number of classes found in the Image dataset.

The softmax layer, the last layer, calculates the probability distribution regarding a set of integers included in an input vector. The softmax function can be defined as an activation function. A vector that includes a set of values indicates the probability of a class or event due to the softmax activation function. The vector's values add up to one. An enormous network's primary issue is that it experiences vanishing gradient descent. In the case when the update to weights resulting from backpropagation is minimal within the lowest layers due to a modest gradient value, this is known as "vanishing gradient descent." Throughout the training, the network simply stops learning. The third (Inception 4 [a]) and sixth (Inception 4 [d]) layers of the architecture, which are intermediary, are given auxiliary classifiers. Only used throughout training, auxiliary classifiers are eliminated during inference. An auxiliary classifier's task is to do classification depending on inputs in the middle of the network and add the training-related loss back to the network's overall loss.

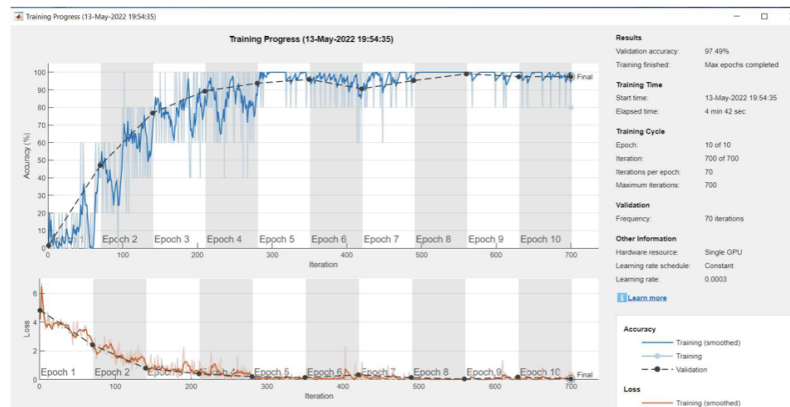


Figure 5: 50-Person Epoch 10 training

V. EXPERIMENTAL RESULTS

Epoch represents the main method to help the model find a way to represent the samples within the dataset with less error; choosing the right number of the epoch is a critical task within the training of the dataset and to compare the difference effecting of a different number of epochs and their connection to the number of persons within the database. The following tables and figure show the effects of increasing number of the epochs+ people samples within the data set which results that the number of epochs when increased and results Overfitting of the model. When using the dynamic number of epochs, this will lead to a lot of times where the weight changes and the training curves go from underfitting to overfitting. By trial and error, not only the number of epochs may affect the accuracy of the system but also the amount of the data as well. According to the results that have been obtained from the training data, 100 persons with ten epochs provided the highest validation accuracy, 98.37%. According to Table II and as shown in figures (5, 6, 7, and 8), with 50 people and 6 min 25 sec of elapsed time, 70 iterations per epoch, and a 0.003 learning rate, a validation accuracy of 98.12% is achieved. With 100 people and 13min 24sec elapsed time with 140 iterations per epoch and 0.003 learning rate results in validation accuracy 98.37%, With 150 people and 14 min 19 sec elapsed time, 210 iterations per epoch, and a 0.003 learning rate, the validation accuracy is 96.25. With 200 people and 17min 2sec elapsed time with 280 iterations per epoch and 0.003 learning rate results in validation accuracy 95.68%.

TABLE III
Training With Ten Epochs and N Number of Samples.

No. of Person	Validation accuracy	Elapsed time	Iterations per epoch	Max iteration	Validation Frequency	Learning rate
50	98.12%	6 min25sec	70	700	70	0.0003
100	98.37%	13 min24sec	140	1400	140	0.0003
150	96.25	14 min19sec	210	2100	210	0.0003
200	95.68%	17 min 2sec	280	2800	280	0.0003

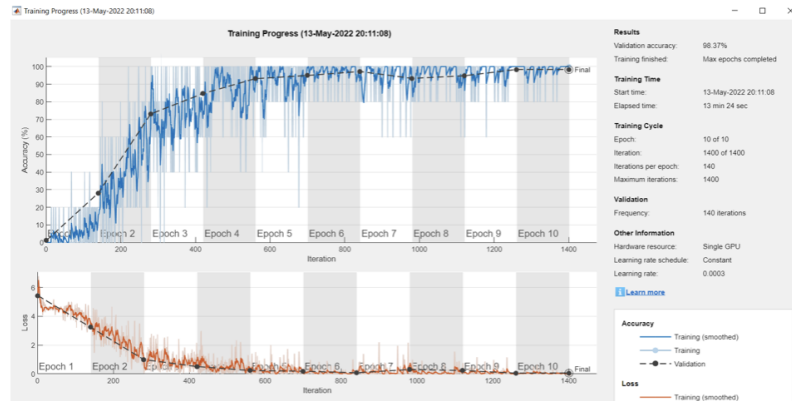


Figure 6: 100-Person Epoch 10 training

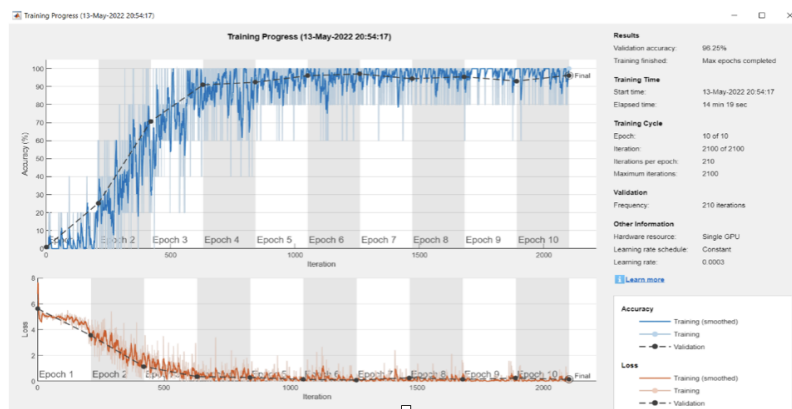


Figure 7: 150 Person Epoch 10 training

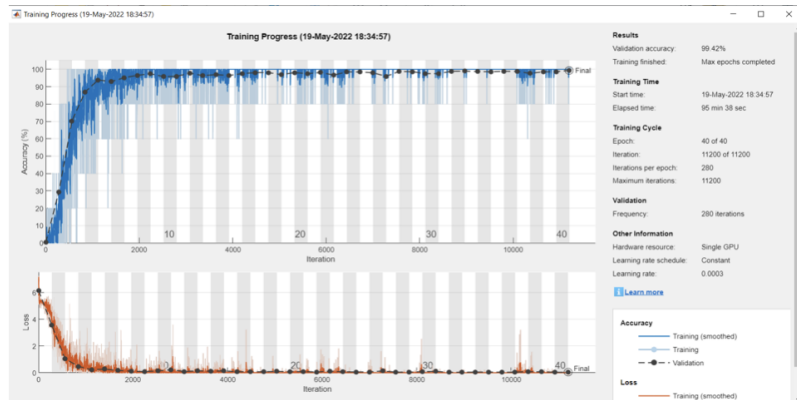


Figure 8: 200 Person Epoch 10 training

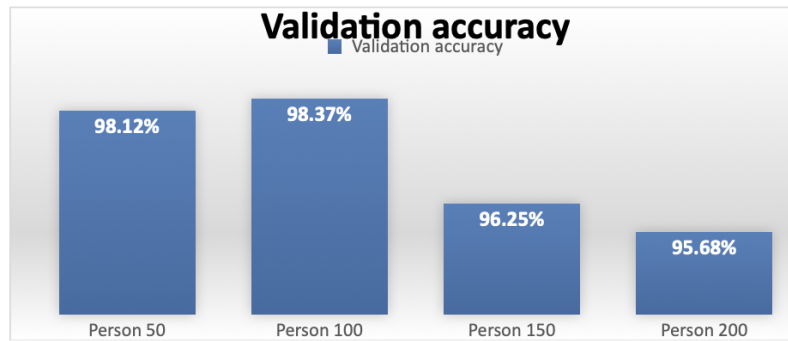


Figure 9: validation accuracy of data 10 epochs in different samples

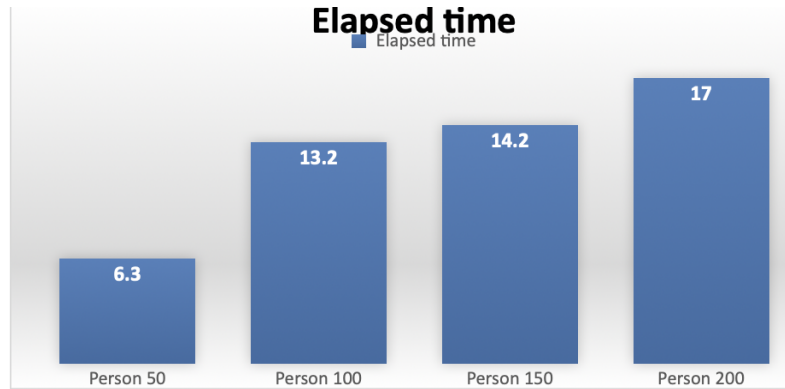


Figure 10: Elapsed time of data 10 epochs in different samples

This matching accuracy can be used with proper Internet of Things techniques to give access authorization to individuals and can be physically applied in intrusion detection systems where the system gives access rights only to those authorized and detects the ones who are not, as well as detecting the intruders. Table III shows the execution of samples within the system database with epochs of 10 and 100.

Many factors may affect the matching accuracy where the system fails to classify some images since it uses background elements and the shape of the person rather than using facial features only. Using a white-label background may also help the process. Balancing the training elapsed time and training accuracy, in this work an epoch of 10 has been chosen and the number of samples (i.e., the persons) is 100. Since the accuracy that has been obtained is high and the time consumption is much lower than the other data, as shown in the figures and tables compared above, the number of epochs and the number of samples is not static within this work. It can be modified if the system administrator decides that time consumption is not the biggest issue. Table IV shows other impact factors that may or may not affect the classification accuracy.

A. Dataset Used

The data set used in this work consists of two different groups, which are:

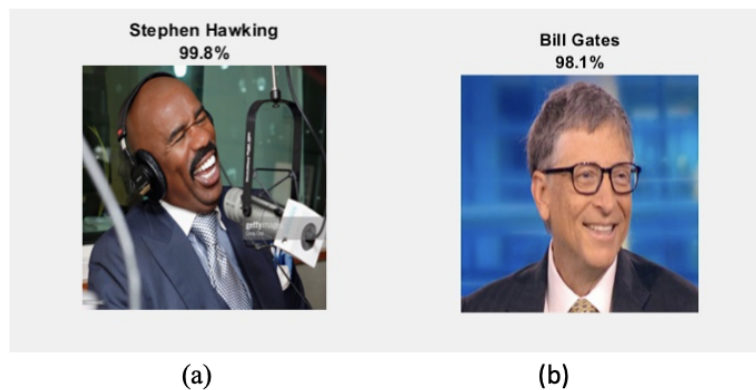


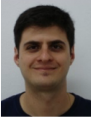







Figure 11: a) incorrectly classified the image b) correctly classified the image

TABLE IV
execution of samples within the system database with epochs of 10 and 100 persons

Image test	Name	True or False	Accuracy	Image training database	Notes
	Bill Gates	true	98.1%		
	S1	true	100%		one color background
	Stephen Hawking	false Steve Harvey	99.8%		Difficulty isolating devices
	Stephen Hawking	true	100%		

- Collected dataset that is applied to known Middle Eastern figures; the core issue of this thesis is the use of deep learning to detect and recognize faces. The primary problem of deep learning is how to train data, so we need to prepare training data and mark the classification of faces in each image. Because it is difficult to find public datasets that meet our requirements, data were collected ourselves. Since it mainly investigates the influence of face proportion on confidence and accuracy, a proportion of face images is the data that needs to be collected.

TABLE V
 other impact factors that affect the classification accuracy

Factor	Discussion
Illumination	Since the image or frame was preprocessed before the processing was applied, the illumination is not a factor impacting the classification negatively or positively
One-color background	The color of the background, if it is one color, may affect the classification accuracy positively since the number of features obtained from the background is much less than complicated background
Scaling and rotation	CNN generates features that are invariant to scale or rotate off the face (within acceptable levels)

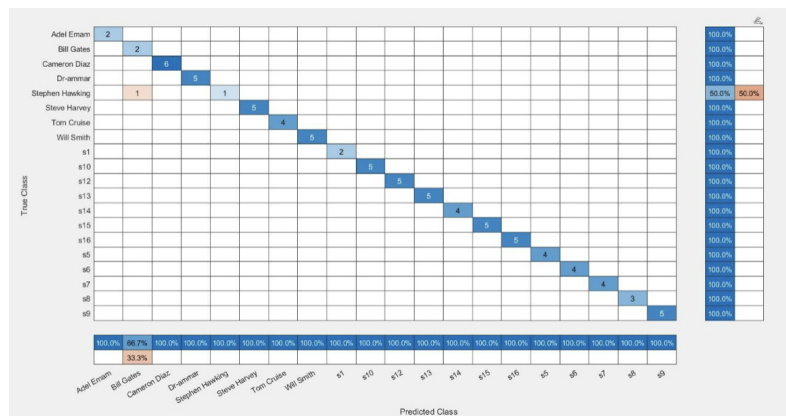


Figure 12: confsion matrix

- The global dataset that the previous author used to recognize faces consists of 200 images that are grouped into four different groups with different snapshots of the people (different ages and genders) at 180 degrees.

B. Performance Analysis

In order to evaluate the system's accuracy and measure the effects of using this data, not only the accuracy was calculated but also the confusion matrix, as shown in Fig. 12 using standard equations and abbreviations in equations 1 and 2 , where the average accuracy of the work has been calculated at 98%, and Fig. 13 represents a snapshot of execution where the detection of Bill Gates was wrong. The system does not approve, it while other detections were approved and considered valid.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$\text{Sensitivity} = TP / (TP + FN) \quad (2)$$

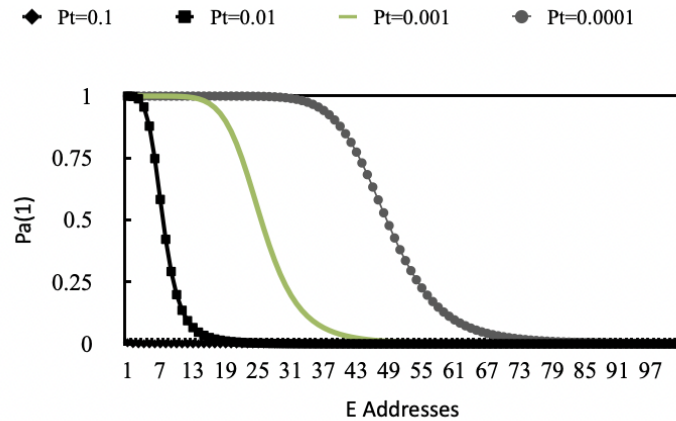


Figure 13: True detection of the images

VI. CONCLUSION

In this research, a system for face detection and identification was proposed using Google-Net architecture that uses transfer learning to allow the network to learn new images based on the learned experience of trained architecture. The design of a network leads to an architecture that maximizes the system's accuracy, accurately detects faces that are saved to the database, and specifies the effects of the weights that were used within the nodes of the hidden layer, which is considered the most time-consuming task within the architecture. Epoch is the primary tool used by the model to find a way to represent the samples in the dataset with the least amount of error. Selecting the appropriate number of epochs during the dataset training is crucial. In this paper, the number of epochs is 10 and the number of training samples is 100 to balance time consumption and accuracy. In the present research, factors like illuminations handled by the convolutional neural network and using one-color background images positively affect the classification accuracy since the number of features that have been obtained from the image is strongly connected to the classified image. Using deep learning as a classification model made the system invariant to rotation and scaling when classifying the object (which is a human face in our case study), which may give more power to the classification process.

Funding

None

ACKNOWLEDGEMENT

The author would like to thank the reviewers for their valuable contribution in the publication of this paper.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

- [1] Q. Hong, Z. Wang, Z. He, N. Wang, X. Tian and T. Lu, "Masked Face Recognition with Identification Association," 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), 2020, pp. 731-735, doi: 10.1109/ICTAI50040.2020.00116.
- [2] Taigman, Yaniv, and Ming Yang. "Systems and methods for facial representation." U.S. Patent 10,095,917, issued October 9, 2018.
- [3] Sun, Yi, Ding Liang, Xiaogang Wang, and Xiaoou Tang. "Deepid3: Face recognition with very deep neural networks." arXiv preprint arXiv:1502.00873 (2015).
- [4] Y. Sun, X. Wang and X. Tang, "Sparsifying Neural Network Connections for Face Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4856-4864, doi: 10.1109/CVPR.2016.525.
- [5] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.
- [6] Qi, Xianbiao, and Lei Zhang. "Face recognition via centralized coordinate learning." arXiv preprint arXiv:1801.05678 (2018).
- [7] Wang, Pin, En Fan, and Peng Wang. "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning." Pattern Recognition Letters 141 (2021): 61-67.
- [8] H. Du, H. Shi, Y. Liu, D. Zeng and T. Mei, "Towards NIR-VIS Masked Face Recognition," in IEEE Signal Processing Letters, vol. 28, pp. 768-772, 2021, doi: 10.1109/LSP.2021.3071663.
- [9] Li, Zheng, Xuemei Lei, and Shuang Liu. "A lightweight deep learning model for cattle face recognition." Computers and Electronics in Agriculture 195 (2022): 106848.
- [10] R. L. Priya, S. Nanda, M. Dandekar, A. Devnani and R. Kadakoti, "Face-Recognition Based Attendance System using One-Shot Learning," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 01-07, doi: 10.1109/ICCCNT51525.2021.9579899.
- [11] Nair, Sreelu P., K. Abhinav Reddy, Prithvi Krishna Alluri, and S. Lalitha. "Face recognition and tracking for security surveillance." Journal of Intelligent & Fuzzy Systems 41, no. 5 (2021): 5337 – 5345.
- [12] F. A. Uckun, H. Azer, E. Nurbaa? and E. Onat, "Direction Finding Using Convolutional Neural Networks and Convolutional Recurrent Neural Networks," 2020 28th Signal Processing and Communications Applications Conference (SIU), 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302448.
- [13] G. Lou and H. Shi, "Face image recognition based on convolutional neural network," in China Communications, vol. 17, no. 2, pp. 117-124, Feb. 2020, doi: 10.23919/JCC.2020.02.010.
- [14] Y. A. Bachtiar and T. Adiono, "Convolutional Neural Network and Maxpooling Architecture on Zynq SoC FPGA," 2019 International Symposium on Electronics and Smart Devices (ISESD), 2019, pp. 1-5, doi: 10.1109/ISESD.2019.8909510.
- [15] Y. Zhang, H. Li, H. Dong, Z. Dai, X. Chen and Z. Li, "Transfer learning algorithm design for feature transfer problem in motor imagery brain-computer interface," in China Communications, vol. 19, no. 2, pp. 39-46, Feb. 2022, doi: 10.23919/JCC.2022.02.004.
- [16] T. A. Jaber, "Artificial intelligence in computer networks," Periodicals of Engineering and Natural Sciences (PEN), vol. 10, no. 1, pp. 309-322, 2022.
- [17] T. A. Jaber and M. A. Hussein, "Study on known models of NB-IoT Applications in Iraqi environments," in IOP Conference Series: Materials Science and Engineering, 2019, vol. 518, no. 5: IOP Publishing, p. 052013.
- [18] T. A. Jaber, "Security Risks of the Metaverse World," International Journal of Interactive Mobile Technologies, vol. 16, no. 13, 2022.
- [19] Tarish, Hiba A., Alaa Q. Rahima, and Tanya A. Jaber. "Proposed collision avoidance system in driverless cars." Engineering and Technology Journal 37, no. 1 (2019): 1-5.